Controls

Design, Commissioning and Operational Experience with the SPEAR-3 Orbit Feedback System

Till Straumann

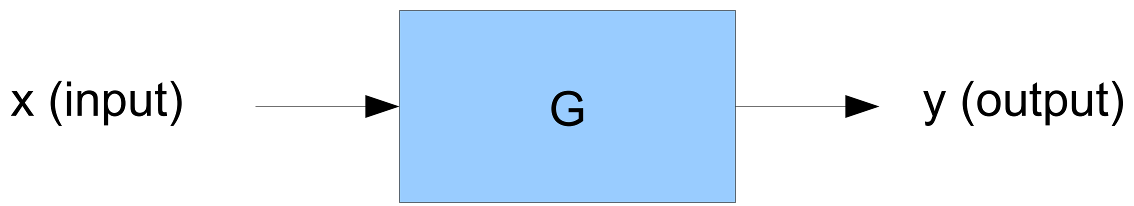
May 9, 2013

Overview

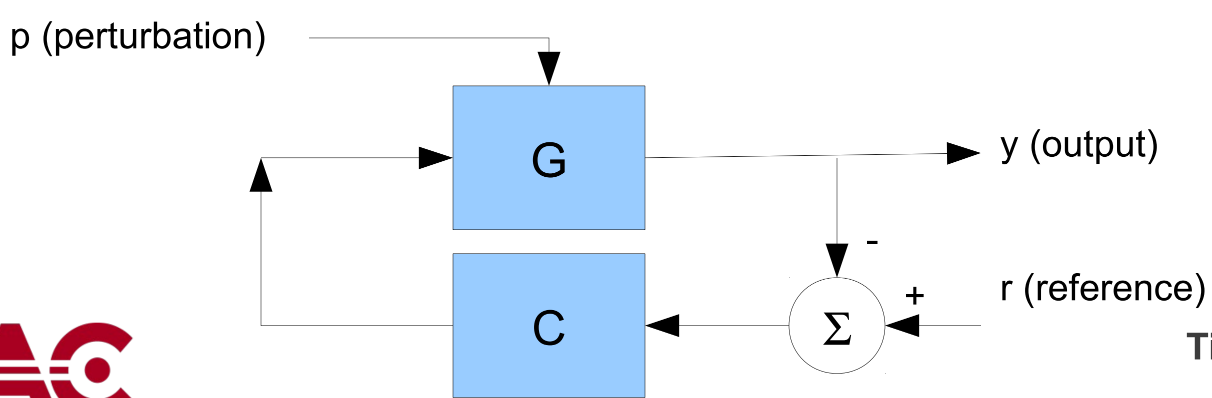
- Recap of elementary control theory/feedback systems
- Design of the SPEAR-3 FOFB
- Commissioning
- Operational experience
- Lessons learned
- Conclusion

Feedback System

- System G ('plant')
  - Has inputs which can manipulate G's internal state
    - However: available inputs may be insufficient to control complete internal state.
  - Has outputs at which (parts of) internal state may be observed
    - However: available outputs may not permit observing complete internal state.



- Feedback: Feed output signal(s) into a controller 'C' which computes a steering signal so that closed loop meets certain criteria.



Design Goals

- Output tracks reference (steady-state, dynamic behavior)
- Output suppresses disturbance (steady-state, dynamic behavior)
- Stability (bounded input → bounded output)
- Keep control signal (X) within bounds, limited slew-rate
- Handle variations of system parameters
- Deal with limited knowledge of G
- Deal with limitations (observability, controllability, dynamics)

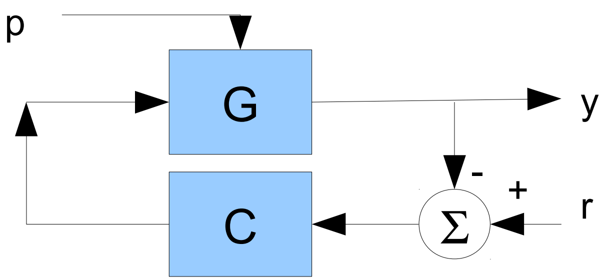
System Analysis & Design

- Assumption: linear, time-invariant (LTI) systems
  - Can be dealt with analytically in many cases
  - Familiar concepts: frequency-domain, fourier/laplace-transform etc.
  - Huge amount of literature
- Justification
  - LTI often reasonable approximation
  - Especially when dealing with small deviations from operating point/steady-state
- However, in some cases one must trespass into domain of non-linear systems. Often only accessible to numerical techniques.

LTI Feedback Loop

- LTI system can be analyzed using transfer functions (fourier-, laplace-, z-transform). Basic loop can be stated as

$$y = \frac{G_1 C r + G_2 p}{1 + G_1 C}$$



- $G_1 C$ 'Loop Gain' determines
  - Tracking (→ 1 for infinite gain)
  - Suppression of disturbance (→ 0 for infinite gain)
  - Stability (→ problem with increasing gain)
  - Dynamic behavior (poles/zeros of C affect poles/zeros of closed loop)

Controller Design

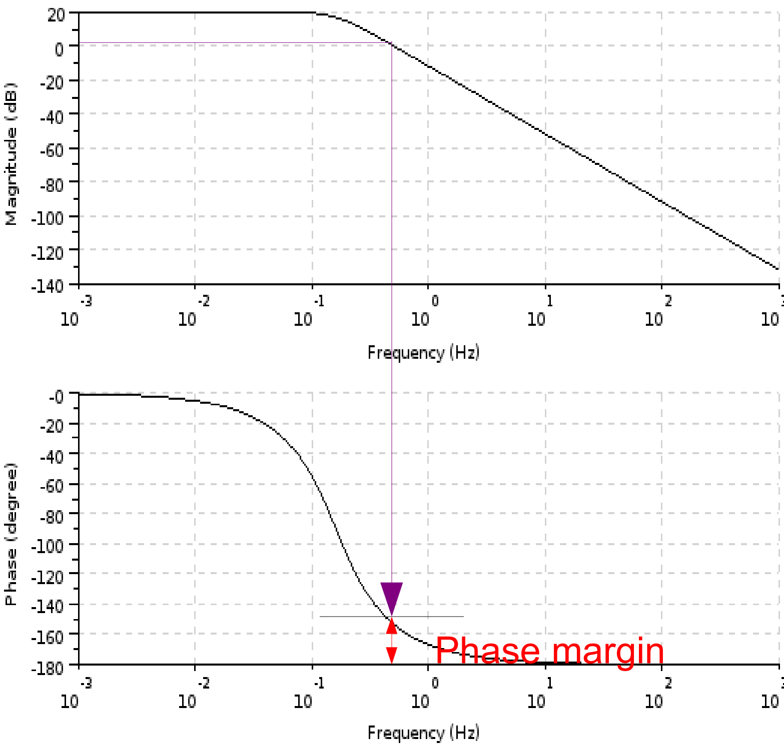
- Behavior of closed loop can be inferred from behavior of open loop (if stable) and controller.
- Controller gain, dynamics (zeros, poles) chosen so that closed-loop behavior optimal in some sense.
  - More advanced controllers (state-space, IMC) offer more flexibility than simple PID.

Stability

- Make loop gain (over frequency) big while keeping closed loop stable.
- Loop gain is *complex*; must not become -1.
- Classical tools: Root-locus plot, Nyquist-, Bode-diagrams
  - E.g., Bode diagram of complex loop gain vs. frequency visualizes stability margin.
    At the frequency where magnitude is unity the phase lag must be less than 180deg.

Bode Diagram

- Bode plot for typical second-order system.
- Desirable closed-loop behavior in frequency areas (→ closed-loop bandwidth) where loop gain >> 1
- However, if gain is increased (uniformly, for all frequencies) phase margin is reduced (bringing system closer to instability)
- Must design response (frequency-dependent 'gain') of controller so that loop-gain and closed-loop bandwidth are optimized while maintaining phase-margin.



Controller Design Easy?

- Can we just synthesize the complex transfer function of the controller so that we obtain any desired gain and phase vs. frequency?

  → *NO*!

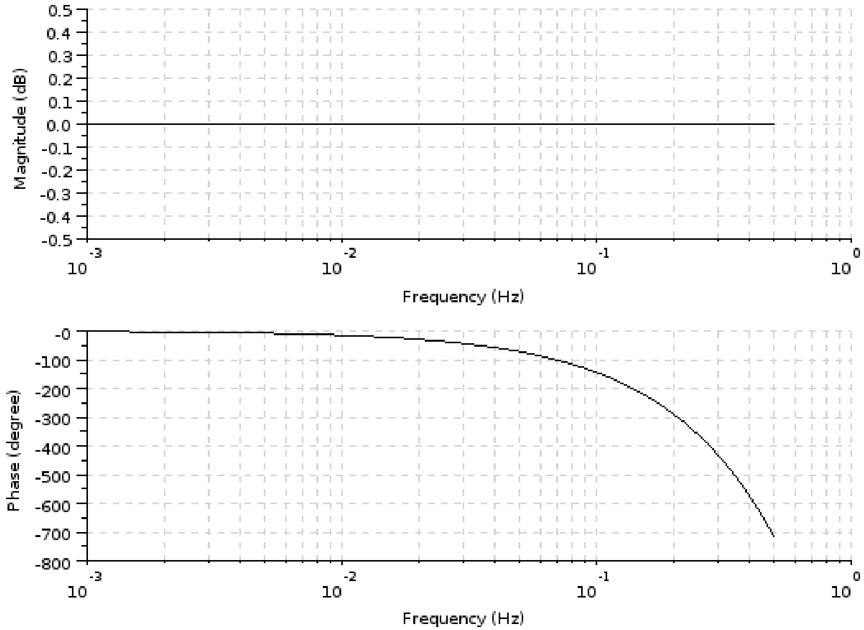
  - Amplitude and phase-response (or real- and imaginary parts) are not independent [Paley-Wiener; Hilbert] ! *Causality* dictates that the phase of a minimal-phase system (more phase lag can be added but is usually not beneficial) can be computed from the amplitude.
  - Causality imposes further restrictions on amplitude response.
  - Additional phase lag is usually bad.

Dynamics of Storage Ring

- Relatively simple; speed limited by
  - Response of correctors + vacuum chamber
  - Power-supplies
  - Dead-time in loop (total propagation delay from taking BPM readings to setting correctors)
- Can often be modeled by low-order system and dead-time.
- Non-linear effects due to limited large-signal performance of power-supplies.

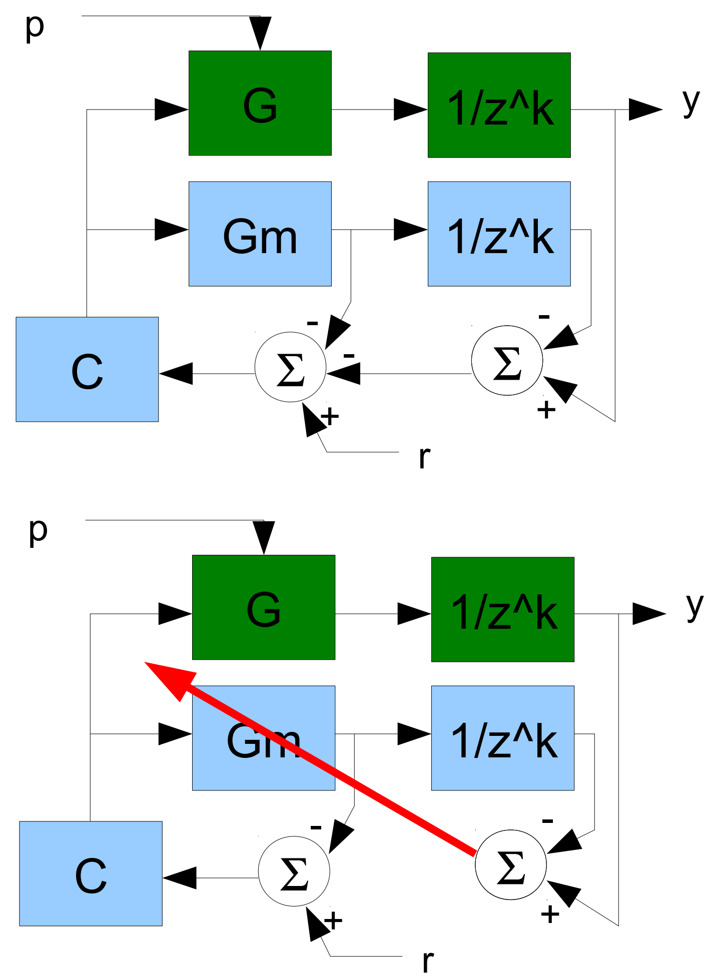
Effect of Dead-Time

- Time delay is an all-pass with fourier-transform $e^{-j\omega T}$ = linear phase (exponential in log-scale of bode plot)
- Phase lag increases rapidly as f>1/(2T)
- *Total dead-time* in the system is critical
  - In a discrete-time system *dead-time* (=total delay in feedback loop) *may be multiple clock cycles*!
- Affects
  - Stability of closed-loop
  - Bandwidth of closed-loop



Mitigation of Dead-Time

- Use delayed output of *model* Gm.
- Observe difference between true output and delayed model
  - Use as input to feed back: "Smith-predictor"
  - Use as driver to improve model: Adaptive filter
- Drive controller with direct model output



SPEAR-3 FOFB

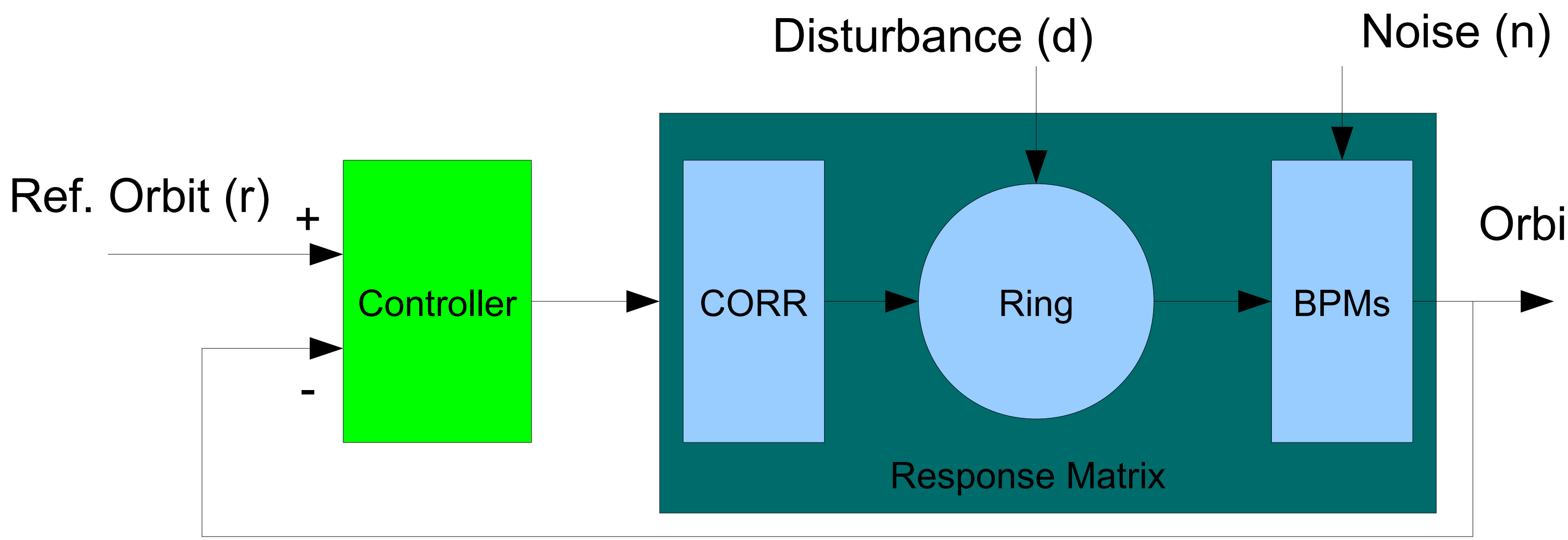
- Early concept dates back to 2000
- Based on COTS components
- Commissioning started in 2005
- Added RF feedback in 2009

SPEAR-3

- 59 Bergoz BPM electronics (analog, multiplexing)
- 108 corrector power supplies (nominally 4kHz small-signal bandwidth).
- Copper vacuum chamber with CuNi inlays for increasing bandwidth of field penetration (~120Hz)
- Submicron orbit stability desired (34mm x 86mm vacuum chamber)

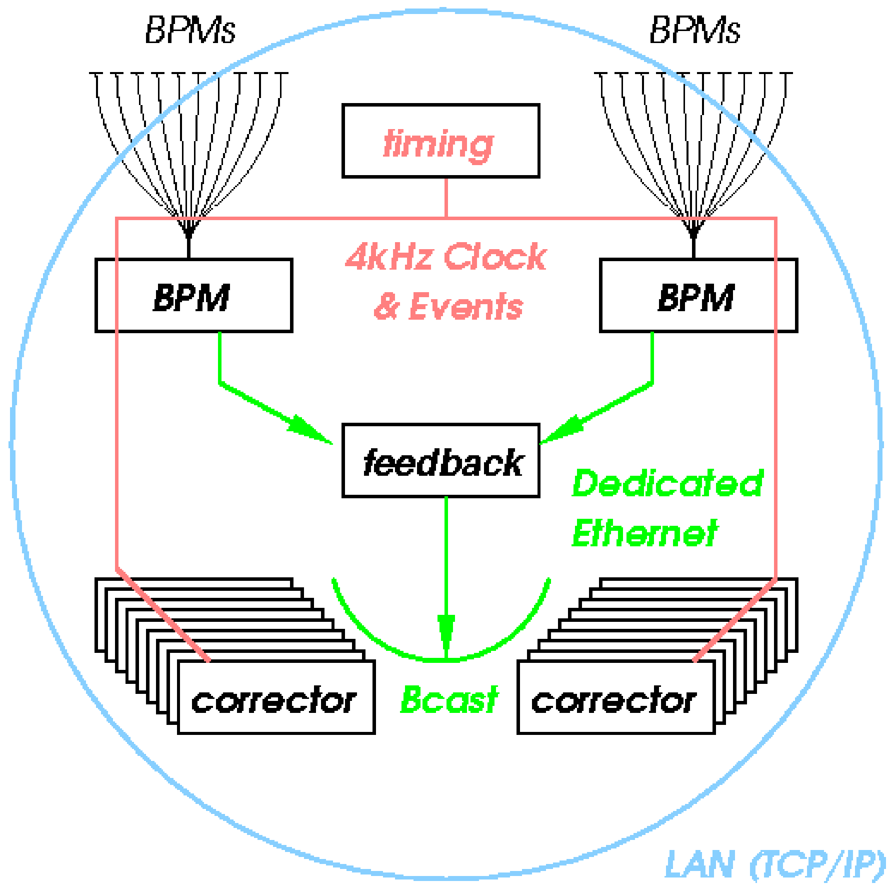
System Model

- Ring characterized by “response matrix”: BPM readings (y) as a function of corrector currents (x) is described as a matrix multiplication: y] = [R] x]
- Beam itself is fast. Dynamics dominated by magnetic field penetration + power supplies
- BPM readings are not instantaneous but ~1-2 orders of magnitude faster
- Note: no way to distinguish BPM noise from 'true' disturbance
  - → Feedback only as good as BPMs



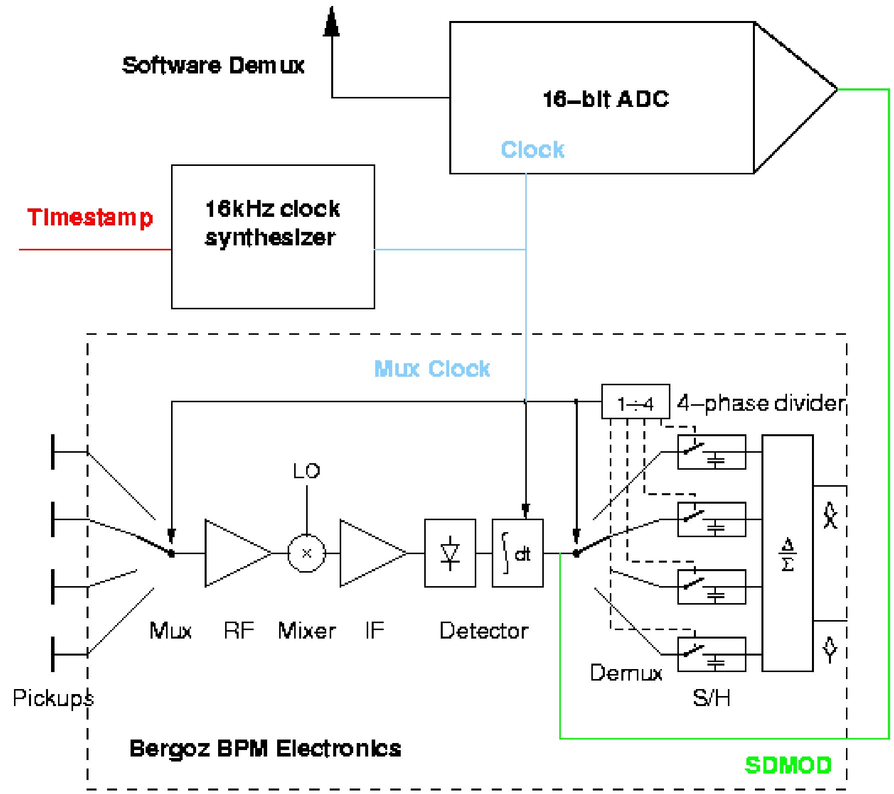
Hardware Architecture

- Early concept called for COTS components
  - Bergoz (analog) BPMs
  - BPM acquisition in two locations
  - Central FOFB CPU
  - In-house designed, intelligent corrector PS controllers
  - 4kHz clock frequency; clock signal (RF subharmonic) globally distributed
  - Data communication (BPM → FOFB → CORR) via dedicated fast (100Mbit) Ethernet.



Bergoz BPM Electronics

- Acquisition of multiplexed base-band signal with a single ADC. Software demux, delta/sum.
- 16kHz sampling; potential for aliasing
- Integrated noise 0-1Hz: ~0.05-0.1um
- Signals of ~32 BPMs acquired by single CPU.



FOFB CPU

- MVME6100 with 1GHz PPC. AltiVec does 116x240 matrix by 240 vector multiplication in ~100us.

Till Straumann
May 9th, 2013

Corrector Power Supplies

- In-house designed power-supply controller with integrated DAC and intelligence (diagnostics). 8 PS in custom crate, controlled by COTS VME CPU board.
- Legacy design. Reuse crate, form-factor and parts of electronics from PEP-II.
- Crate deeper than VME. 'Franken(stein)' board with FPGA mimicks VME signals to CPU. Contains MMIO 'registers' which control 8 PS over backplane.

Dedicated Network

- COTS Fast Ethernet (100Mbit/s)
- Two links from BPM processors to FOFB CPU
- Link to repeater which fans out to PS controller CPUs. Use ethernet broadcast.
- Dedicated network; no other traffic. Achieve determinism.

Timing

- No sophisticated timing system was planned or budgeted.
- Synchronous (RF subharmonic), global 4kHz clock distribution.
- Simple serial protocol was added to clock signal allowing for distribution of a timestamp (“Cycle ID”) and up to 7 triggers.

Till Straumann
May 9th, 2013

Basic Software Architecture

Controls

- Use EPICS for slow controls + monitoring
- All CPUs/IOCs run hard-real-time OS (RTEMS)
- Real-time controls and diagnostics are non-EPICS and have higher priority.

Real-Time Software

- BPM
  - readings are time-stamped with CID, sent on PtP ethernet.
  - Cycle-by-cycle history buffer which can be triggered via timing system. Supports 'pre-trigger' (continuously running ring-buffer which is frozen at or after trigger).
- Communication
  - CID 'travels' with data (BPM reading, setpoints) for diagnostic purposes
- Correctors
  - setpoints can be taken from dedicated ethernet
  - For diagnostics: waveform table, clocked at 4kHz. Start can be triggered via timing system
- FOFB Controller
  - Can archive orbit data (received via PtP ethernet) into cycle-by-cycle history
  - Can send setpoints in “open-loop mode”

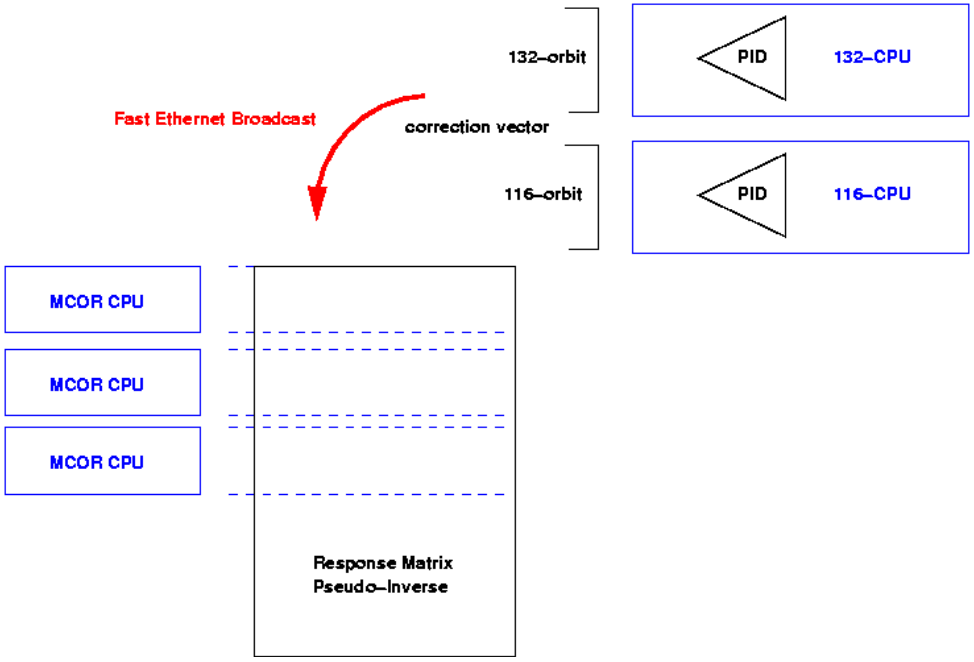
Cycle-by-Cycle Diagnostics

- 4kHz orbit history (e.g., used to compute PSD)
- Characterization of open-loop response (w/o FOFB path)
  - Write e.g., step to corrector waveform table(s)
  - Arm BPM history buffers and setpoint table(s)
  - Send synchronous trigger via timing system
- Characterization of full open-loop path (see later)

FOFB Algorithm (first idea)

Controls

- Costly operation is multiplication of orbit vector by inverse of response-matrix (RI)
- Idea: keep RI matrix 'distributed' in corrector controller. Each one needs only 'its' 8 rows.
- → each PS controller computes only small matrix by vector
- PID algorithm on orbit error
- Observations
  - Noise
  - Instabilities
  - High corrector currents would build up
- *Why doesn't this work?*



Ill-Conditioned System!

$$y]=[R]x]$$
$$x]=[RI]PID(r]-y])$$

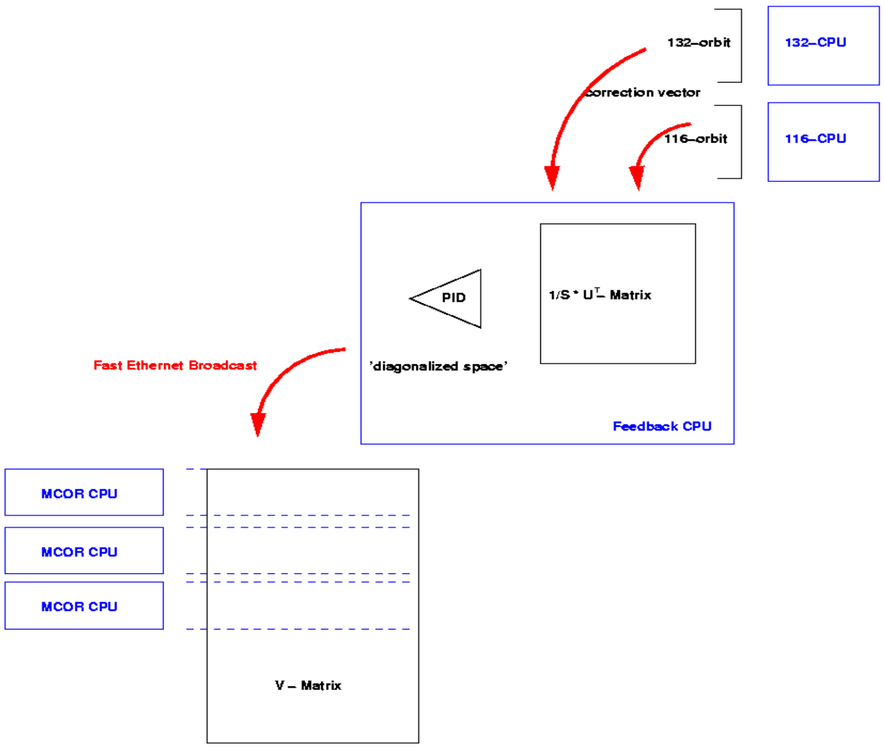
- Look at integrator only: $PID(u]) = Diag(1/s)u] = 1/s[1]u]$

$$(s[1]-[R][RI])y] = [R][RI]x]$$

- Eigenvalues of [R][RI] must be in left half-plane!
- However, RI – which is computed employing the SVD technique uses less singular values than the smaller of R's dimensions
- → Must only have as many integrators in the system as there are significant singular values!

Revised Algorithm

- FOFB CPU projects orbit into `eigenspace' $[1/\sigma][U]^T$
- Run as many PIDs as there are significant singular values (a loop for each `mode').
- Send out vector of `modal' corrections
- Corrector projects modal corrections into corrector current using row of $[V]$



Commissioning

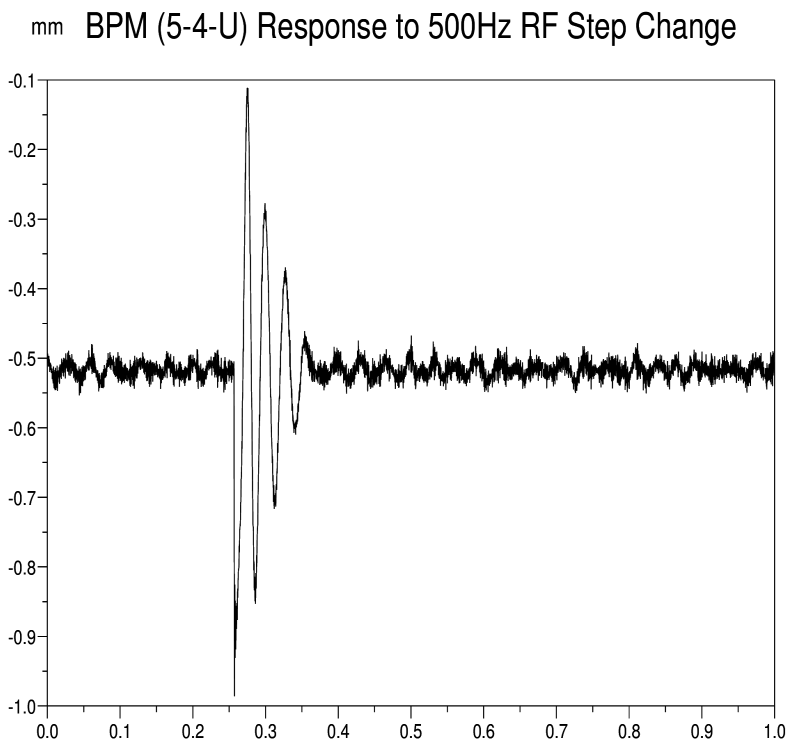
- All operating parameters programmable via EPICS
  - Response matrix inverse part $[1/\sigma][U]^T$
  - Rows of [V] matrix
  - PI coefficient vectors Ki, Kp (one element for each mode)
  - Target orbit: 2 setpoints, 'golden' and 'delta'
  - Start/stop
  - Trip limits
    - If orbit error grows too big
    - If modal corrections grow too big
    - Other errors (e.g., ethernet link failure, bad BPMs)

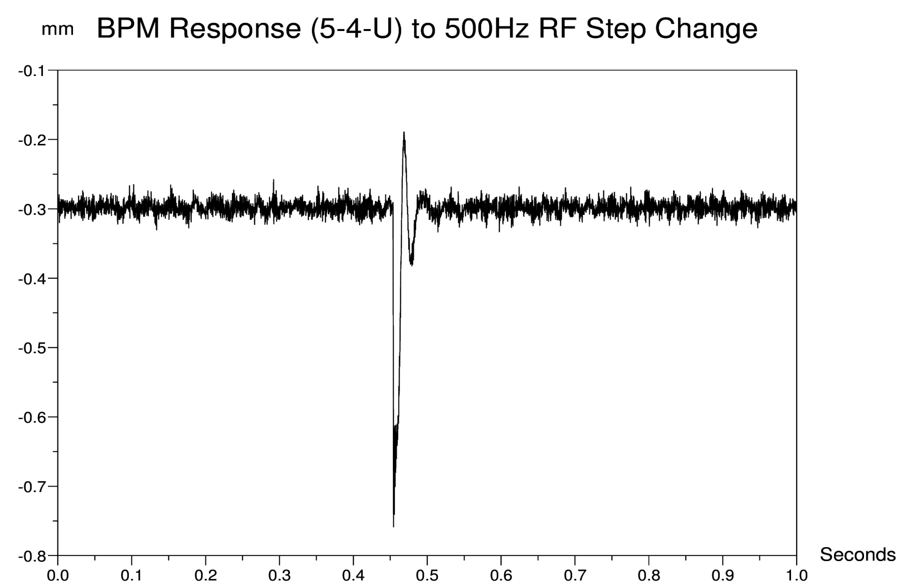
Tuning

- Most work done by physicists
- Extensive use of matlab (with EPICS/CA interface)
- Workflow
  - Take 4kHz orbit data using 4kHz history buffers. Either for steady state or synchronously apply small perturbation (target orbit; step RF).
  - Analyze data in matlab
  - Tune Ki/Kp

Tuning: Response to Step of RF

Controls





SLAC NATIONAL ACCELERATOR LABORATORY

Till Straumann
May 9th, 2013

Typical Performance

Controls

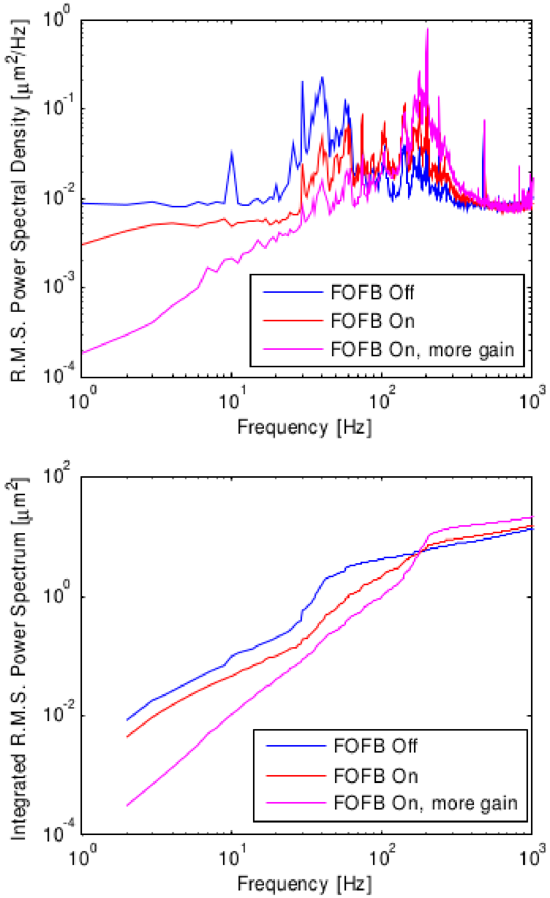


Figure 3: Orbit error (spacial r.m.s.) power spectrum and power spectral density

- From 2006 EPAC paper (THPCH102)
- Currently: Stability limited by
  - BPM noise/inaccuracies
  - Ground motion

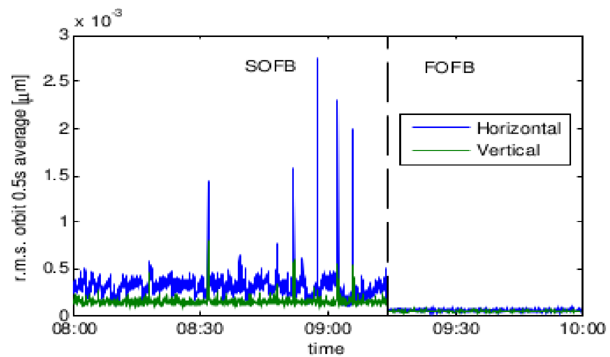


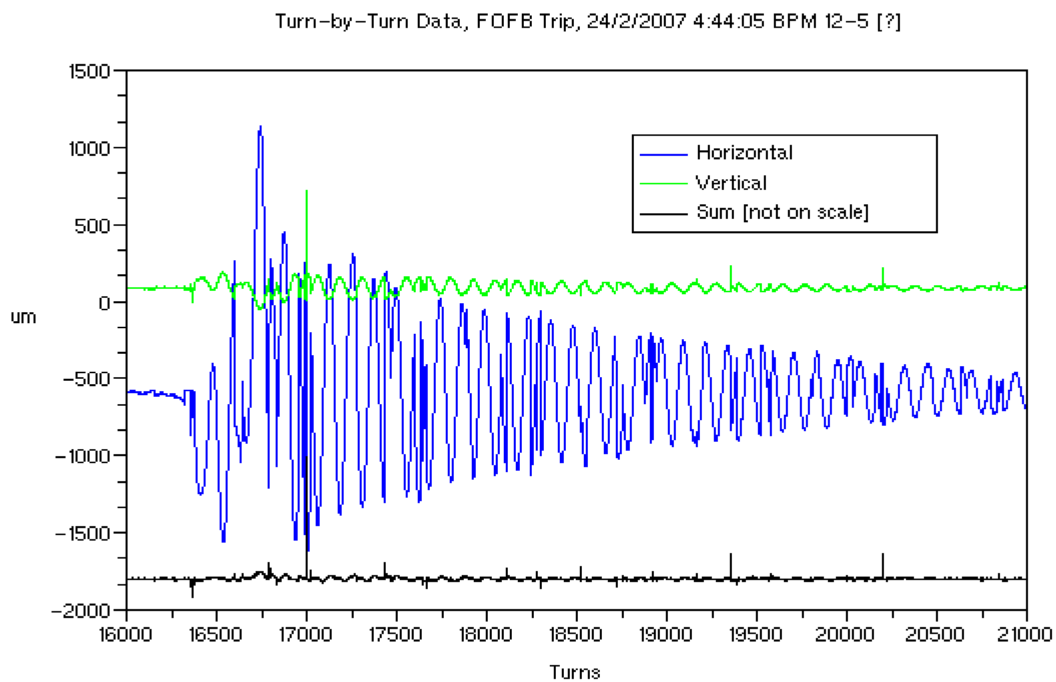
Figure 4: Suppression of orbit errors due to undulator gap changes and other localized disturbances.

Operational Experience

- After commissioning the operators have been given a 'one button' interface where they can start/stop FOFB and where faults are flagged.
- Not many incidents are reported to me.
- Some examples of what we did after FOFB had long been commissioned:
  - Mysterious trips
  - RF feedback
  - Characterization of open-loop response

Mysterious Trips

- Sometimes, FOFB would just trip with 'orbit violation' (orbit too far from target).
- 4kHz orbit history buffer triggered on violation: reported large excursions.
- Data taken with fast, digital turn-by-turn BPM revealed actual excursions.
- Could be tracked down to sparks in waveguide.
- *Physicists are not engineers (unethical @!*)*: implemented 'glitch filter' which tolerates short bursts of orb



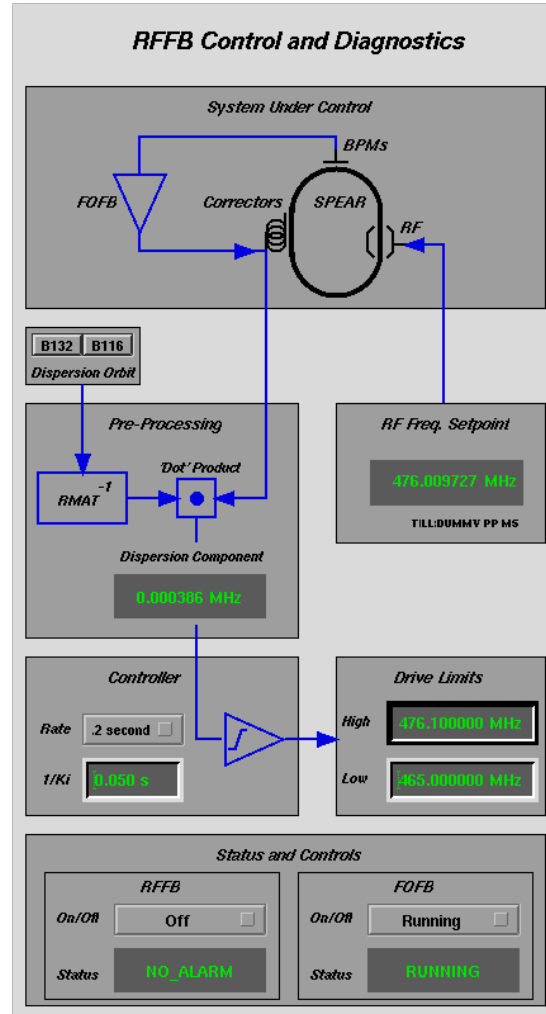
RF Feedback

Controls

- FOFB tends to try to correct dispersion; buildup of corrector currents.
- RF Feedback (formerly a standalone app) is now integrated with FOFB
- FOFB monitors dispersion component 'd' in modal setpoint vector sm

$$d = dispersionorbit]^T [U][1/\sigma] sm]^T$$

- Slow EPICS feedback drives this component to zero by tuning RF



SLAC NATIONAL ACCELERATOR LABORATORY

Till Straumann
May 9th, 2013

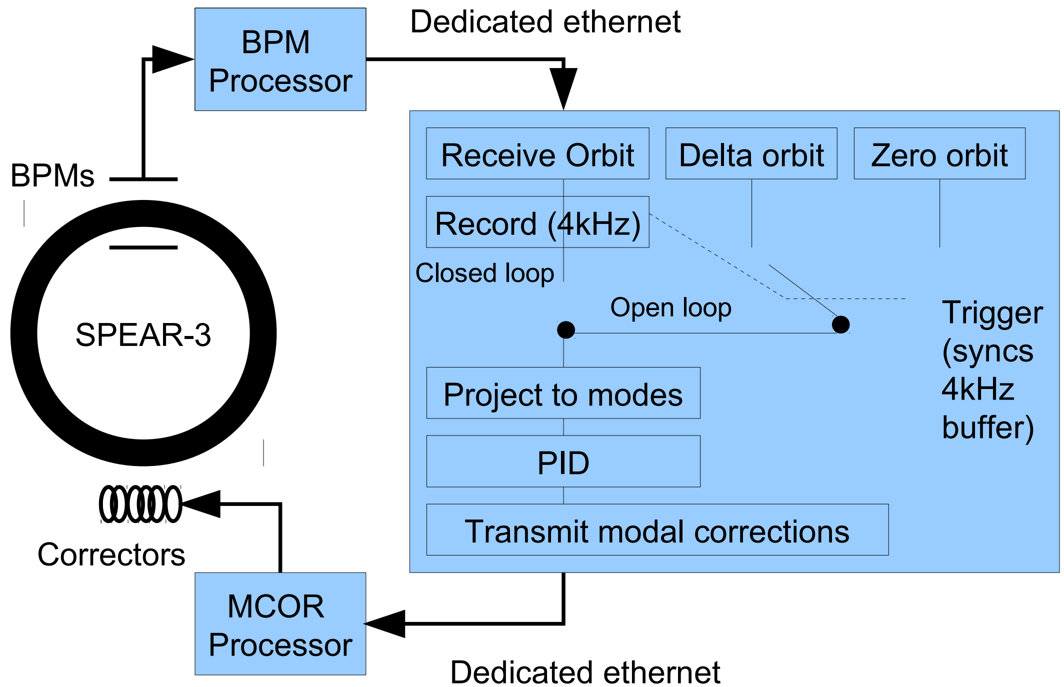
Open Loop Response

Controls

- Characterize FOFB in open-loop mode in order to tune coefficients; improve closed-loop behavior.
- Measure open-loop step-response for individual “modes”.

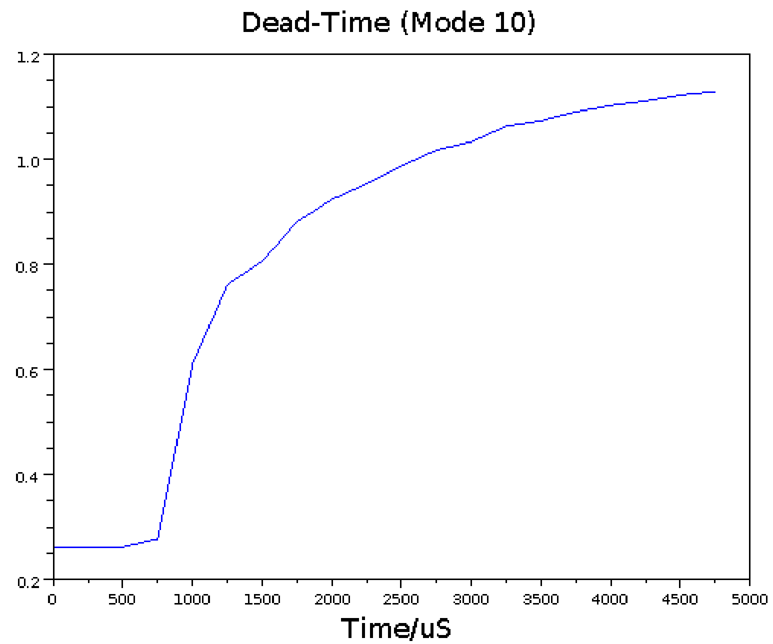
Measurement Setup

- Received orbits are recorded into 4kHz history buffer.
- Loop is broken so that arbitrary, static (instead of “real”) orbit is propagated into algorithm.
- Trigger starts 4kHz acquisition simultaneously with switching between two static orbits thus creating a step.
- Full delay through complete system is measured.



Result

- Dominant system parameters:
  - *750us* (=3 cycles) *dead-time*
  - Behaves roughly like a 1st order system with *100Hz cut-off* freq.



Lessons Learned

- Overall approach using COTS, general-purpose hardware (rather than FPGAs) seems suitable.
  - 10y old components can sustain 4kHz clock rate, 100Hz closed-loop BW, O(100x200) response matrix.
  - Modern components are >10 times more powerful. Should handle small to medium sized storage ring just fine.
  - Obvious advantages of COTS. Examples
    - Obsolete CPU card → exchange with a new one. Almost w/o software changes; minor task.
    - Increase speed (remember: original vs. modified algorithm required more horsepower): exchange CPU cards, upgrade ethernet to GigE
    - By comparison: the in-house designed+built 'Frankenboard' (FPGA) is now a problem. Running out of spares, cannot build new ones w/o respin due to parts obsolescence.
  - Write software instead of firmware. E.g., easy to add new diagnostic tools.

Lessons Learned (cont.)

Controls

- Pay more attention to dead-time; compensation could be added to existing system.
- Better BPM electronics (the system actually supports a mix of Bergoz-, digital- and photon BPMs – the latter are just not used; I don't know exactly why...)
- Implement a simulator! Having to do most studies + tuning on the real system is very time-consuming (schedule shift, beam-loss etc.).
  - Off-line simulator which implements algorithms and models; e.g., in matlab.
  - On-line simulator which simulates correctors, ring and BPMs but hooks into the real software so that as much of the real system can be tested (including time-budget etc.) w/o using the machine. Especially easy with presented system: hook into dedicated ethernet. Let simulator listen to setpoint broadcast and feed simulated BPM readings back.
- Physicists and controls engineers use different names, ordering of BPM vectors, format of response matrix etc. Mapping back and forth is painful and error-prone. Closest contact between the two “worlds” is during precious beam-time when we work together. Loss of efficiency.
  - The control-system proper (EPICS) implements only lowest levels. Higher levels are done in matlab, but physicists and engineers have different upper layers. It could be beneficial to integrate some mid-level functionality into the control system (but e.g., EPICS has no 'matrix' – must format everything into a one-dimensional waveform).

Conclusion

- The SPEAR-3 FOFB does its job quite reliably
- Somewhat aged
- General approach still believed to be adequate